

**What is Claimed:**

1. A system for improving the efficiency of a message processing system, comprising:  
a mechanism for using performance metrics to determine the workload of the message processing system and to determine whether the message processing system is overburdened with messages; and  
a mechanism for restricting the number and/or type of new messages, whereby the system is able to help the message processing system to process its existing work if the system determines the message processing system to be overburdened.
2. The system of claim 1, wherein the performance metrics are based on memory consumed by active message processing.
3. The system of claim 1, wherein the performance metrics are based on processing power used for active message processing.
4. The system of claim 1, wherein the system uses the performance metrics to predict how long an idle process will remain idle, thereby enabling a determination as to whether the system should move the process out of active memory and into secondary storage, thereby improving system performance.
5. The system of claim 4, wherein the performance metrics are based an idle time of a past process.
6. The system of claim 4, wherein the performance metrics are assigned according to a predetermined criterion if no performance metrics are available.
7. The system of claim 4, wherein the system determines whether the predicted duration of idle time exceeds a predetermined threshold.

8. The system of claim 7, wherein the predetermined threshold is based on a prior idle process.

9. The system of claim 1, wherein the message processing system is implementing a business method process using distributed application software.

10. A method for improving the efficiency of a message processing system, comprising:  
determining a workload of a message processing system;  
polling for a new message according to the workload status;  
identifying a blocked instance being processed by the message processing system;  
calculating an expected idle time for the blocked instance; and  
dehydrating the blocked instance if the expected idle time exceeds a predetermined threshold.

11. The method of claim 10, wherein determining the workload comprises accessing performance data regarding the message processing system, and determining, using the performance data, the workload with respect to a system operating parameter.

12. The method of claim 11, wherein the accessed performance data is memory usage.

13. The method of claim 11, wherein the accessed performance data is processor power in use by the message processing system.

14. The method of claim 10, further comprising updating the workload according to the dehydration of the instance.

15. The method of claim 10, further comprising updating the threshold according to the workload.

16. The method of claim 10, wherein the polling step is carried out at a frequency that is inversely proportional to the workload.

17. The method of claim 10, wherein the polling step is carried out at one of a first or second frequencies, wherein the first frequency is greater than the second frequency.
18. The method of claim 17, wherein the polling step further comprises polling only for a new non-activation message.
19. The method of claim 10, wherein calculating the expected idle time comprises:
  - accessing performance data for the message processing system;
  - determining a length of time the blocked instance has been idle; and
  - generating the expected idle time based on the performance data and length of time the blocked instance has been idle.
20. The method of claim 19, wherein the performance data is assigned according to a predetermined criterion if no performance data is accessible.
21. The method of claim 10, wherein the blocked instance is a first instance, and the performance data comprises a recorded idle time of a second instance.
22. The method of claim 10, wherein the message processing system is implemented in business process automation software.
23. A method for managing a workload of a message processing system, comprising:
  - determining the workload of the message processing system;
  - polling for a new message at a frequency, wherein the frequency is inversely proportional to the workload and, if the workload is above a predetermined limit, polling only for a new non-activation message;
  - identifying a blocked instance being processed by the message processing system and, if the blocked instance has no executable segments:
    - calculating an expected idle time for the blocked instance based on performance data relating to the message processing system; and

determining whether the expected idle time exceeds a predetermined threshold and, if so, dehydrating the blocked instance;

updating the workload according to the dehydration; and

updating the performance data according to the polling of the new message.

24. The method of claim 23, wherein the new message relates to a business method process.

25. A computer-readable medium having computer-readable instructions for performing a method for improving the efficiency of a message processing system, the method comprising:

determining a workload of a message processing system;

polling for a new message according to the workload;

identifying a blocked instance being processed by the message processing system;

calculating an expected idle time for the blocked instance; and

dehydrating the blocked instance if the expected idle time exceeds a predetermined threshold.

26. The computer-readable medium of claim 25, wherein determining the workload comprises accessing performance data regarding the message processing system, and determining, using the performance data, the workload with respect to a system operating parameter.

27. The computer-readable medium of claim 26, wherein the accessed performance data is memory usage.

28. The computer-readable medium of claim 26, wherein the accessed performance data is processor power in use by the message processing system.

29. The computer-readable medium of claim 25, the method further comprising updating the workload according to the dehydration of the instance.

30. The computer-readable medium of claim 25, the method further comprising updating the threshold according to the workload.

31. The computer-readable medium of claim 25, wherein the polling step is carried out at a frequency, wherein the frequency is inversely proportional to the workload.

32. The computer-readable medium of claim 25, wherein the polling step is carried out at one of a first or second frequencies, wherein the first frequency is greater than the second frequency.

33. The computer-readable medium of claim 32, wherein the polling step further comprises polling only for a new non-activation message.

34. The computer-readable medium of claim 25, wherein calculating the expected idle time comprises:

accessing performance data for the message processing system;  
determining a length of time the blocked instance has been idle; and  
generating the expected idle time based on the performance data and length of time the blocked instance has been idle.

35. The computer-readable medium of claim 34, wherein the performance data is assigned according to a predetermined criterion if no performance data is accessible.

36. The computer-readable medium of claim 25, wherein the blocked instance is a first instance, and the performance data comprises a recorded idle time of a second instance.

37. The computer-readable medium of claim 25, wherein the message processing system is implemented in business process automation software.

38. A computer-readable medium having computer-executable instructions for performing a method for managing a workload of a message processing system, the method comprising:

determining the workload of the message processing system;

polling for a new message at a frequency, wherein the frequency is inversely proportional to the workload and, if the workload is above a predetermined limit, polling only for a new non-activation message;

identifying a blocked instance being processed by the message processing system and, if the blocked instance has no executable segments:

calculating an expected idle time for the blocked instance based on performance data relating to the message processing system; and

determining whether the expected idle time exceeds a predetermined threshold and, if so, dehydrating the blocked instance;

updating the workload according to the dehydration; and

updating the performance data according to the polling of the new message.

39. The computer-readable medium of claim 38, wherein the new message relates to a business method process.